Amendments to the Specification:

Applicant presents replacement paragraphs below indicating the changes with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing.

Please replace the paragraph beginning at page 5 (continuing on page 6), line 29 which starts with "In one embodiment, the model of work is based on" with:

"In one embodiment, the model of work is based on the observation that in most existing database systems, query operators are usually implemented using an iterator model. In the iterator model, each physical query operator in the query execution plan exports a standard interface for query processing. The operators in this interface include OpenO, CloseO and GetNextO calls. Each time a GetNextO operator is issued an item, such as a tuple or group is returned. Referring to Figure [[4]] 3, the work W is modeled 122 as the total number of GetNextO calls issued throughout the query pipeline including the root in one embodiment. The method counts 124 each GetNextO call K as a primitive operation of query processing and models 126 the total work done by the query as the total number N of GetNextO calls. Query progress is then estimated 128 by dividing the GetNextO count by the estimated total number of GetNextO operators."

Please replace the paragraph beginning at page 15 (continuing on page 16), line 22 which starts with "Given a starting node of a pipeline, the pipeline is defined as" with:

Given a starting node of a pipeline, the pipeline is defined as the longest sequence of non-blocking operators from the starting node. Thus all nodes in a pipeline execute together. Of course, the determination of whether or not a node (i.e., physical operator) is blocking depends on the specific operator. For example, an Index Nested Loops join is non-blocking, whereas a Hash Join is blocking. Given an execution plan, the method generates the corresponding set of pipelines by traversing the nodes ofthe tree in post order and accumulating pipelines using multiple stacks. Referring to Figures 5 and 7, given an execution plan, there are two possible kinds of pipelines. Figure 5 shows a first type of pipeline 150 that is a linear chain of nodes in which there is a unique leaf node 152 that is picked as the driver. Referring to Figure 7, another possible pipeline [[170]] comprises multiple input nodes 172 feeding into a single node 174. In the example of Figure 7, this single node 174 Merge node in a Sort-Merge Join. In this case, all the leaf nodes would be considered driver nodes. When the execution plan of the query is not a single pipeline, the driver nodes are not limited to Table Scan, Index Scan and Index Seek operators. The following examples illustrate sOf!1e sample execution plans and the corresponding driver nodes.

Please replace the paragraph beginning at page 16, line 7 which starts with "Figure 8 illustrates dividing a query execution plan 180 into pipelines" with:

"Figure 8 illustrates dividing a query execution plan 180 into pipelines. Assuming that A is the build side of the Hash Join 182 and B is the probe side, the pipelines are: {Table Scan A 184, Filter 186} 188, {Table Scan B 190, Hash Join 192, Index Nested Loops 194, Index Seek C 196} 198. ~~The driver nodes for the respective pipelines are Table~~"

Please replace the paragraph beginning at page 17, line 25 which starts with "Refining an upper or lower bound for a particular node" with:

"Refining an upper or lower bound for a particular node could potentially help refine the upper or lower bound of other nodes above the refined node in the execution tree. For example, in Figure 11, suppose that at some point in time T during the query's execution, it is concluded that ~~conclude that~~ the upper bound for the Hash Join can be reduced from 1 million rows to 0.5 million rows. Suppose the upper bounds for the Group By and Sort nodes were 0.8 million rows. Then, based on the properties of GroupBy and Sort nodes, it can also be concluded that each of their upper bounds cannot exceed 0.5 million rows. The lowering of the upper bound could help refine the estimates Ni at one or both of these nodes at time T. Note that even when estimator PROG uses only the driver node cardinalities for progress estimation, it may be useful to refine cardinalities of all nodes in the plan since these estimates could influence the estimates Ni of the driver nodes above it. In one embodiment, these bounds are propagated up the tree as soon as a change in the bound can be made for some node. In one embodiment, the frequency at which such propagation of refinement is done can be limited to control the overhead imposed by the propagation. For example, the bounds could be propagated a few times per second at roughly the granularity at which feedback is necessary to the user for a given application."

Please replace the paragraph beginning at page 26, line 20 which starts with "Figure 9 and the following discussion" with:

"Figure [[9]] 13 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including handheld devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices."

Please replace the paragraph beginning at page 27, line 3 which starts with "With reference to Figure 9, an exemplary system" with:

"With reference to Figure [[9]] <u>13</u>, an exemplary system for implementing the invention includes a general purpose computing device in the fonn of a conventional personal computer 20, including a processing unit 21, a system memory 22, and a system bus 24 that couples various system components including system memory 22 to processing unit 21. System bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. System memory 22 includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer infonnation between elements within personal computer 20, such as during start-up, is stored in ROM 24. Personal computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29 and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM or other optical media. Hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for personal computer 20. Although the exemplary environment described herein employs a hard disk 27, a removable magnetic disk 29 and a removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer-readable media which can store data that is accessible by computer, such as random access memories (RAMs), read only memories (ROMs), and the like may also be used in the exemplary operating environment."

Please replace the paragraph beginning at page 28, line 9 which starts with "Personal computer 20 may operate in a networked environment" with:

"Personal computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. Remote computer 49 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all ofthe elements described above relative to personal computer 20, although only a memory storage device 50 has been illustrated in Figure [[9]] ~~13~~. The logical connections depicted in Figure [[9]] ~~13~~ include local area network (LAN) 51 and a widearea network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet."